

00862.023356



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: )  
: Examiner: A. Thomas  
KOUCHI SUGIYAMA )  
: Group Art Unit: 2625  
Application No.: 10/728,951 )  
: Confirmation No.: 9003  
Filed: December 8, 2003 )  
:   
For: PRINTING CONTROL )  
METHOD : January 11, 2008

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

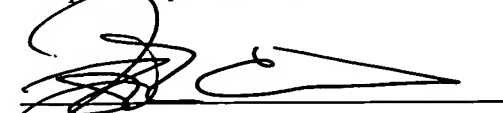
Sir

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed is a certified copy of the following Japanese application:

2002-358313, filed December 10, 2002.

Applicant's undersigned attorney may be reached in our Costa Mesa, CA office at (714) 540-8700. All correspondence should continue to be directed to our below-listed address.

Respectfully submitted,

  
Frank L. Cire  
Attorney for Applicant  
Registration No. 42,419

FITZPATRICK, CELLA, HARPER & SCINTO  
30 Rockefeller Plaza  
New York, New York 10112-3800  
Facsimile: (212) 218-2200

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 2 年 1 2 月 1 0 日  
Date of Application:

出 願 番 号                      特 願 2 0 0 2 - 3 5 8 3 1 3  
Application Number:

パリ条約による外国への出願  
に用いる優先権の主張の基礎  
となる出願の国コードと出願  
号

the country code and number  
of your priority application,  
to be used for filing abroad  
under the Paris Convention, is

J P 2 0 0 2 - 3 5 8 3 1 3

願      人                      キヤノン株式会社  
Applicant(s):

2 0 0 7 年 1 2 月 1 4 日

特許庁長官  
Commissioner,  
Japan Patent Office

肥塚 雅博



【書類名】 特許願

【整理番号】 224584

【提出日】 平成14年12月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明の名称】 印刷制御方法

【請求項の数】 1

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

【氏名】 杉山 晃一

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康德

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100112508

【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100116894

【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

## 【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 印刷制御方法

【特許請求の範囲】

【請求項 1】 バナー印刷の機能と複数のジョブを結合する機能とを有する印刷システムの印刷制御方法であって、

バナー印刷を含む複数の印刷ジョブを結合して所定の加工設定を指定する際に、バナー印刷のデータを複数の印刷データと別に処理する処理工程と、

前記複数の印刷データを 1 つの印刷データに結合する結合工程と、

前記処理工程で処理されたバナー印刷のデータと前記結合された印刷データとを出力する出力工程とを有することを特徴とする印刷制御方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、パーソナルコンピュータなどの情報処理装置とプリンタなどの印刷装置からなる印刷システムにおける印刷制御方法に関するものである。

【0 0 0 2】

【従来の技術】

従来、通常の印刷物の先頭に情報（印刷者名、印刷時間、印刷物名称など）を記載した紙を一枚挿入することにより、印刷物の識別を容易にするバナー印刷という機能がある。また、同様な機能として、印刷データを中間コードに変換した後で一時的に保存し、保存した複数の中間データを 1 つの中間コードに結合する処理を行い、複数のアプリケーションからの印刷に対して一括のレイアウト処理や仕上げ処理を可能にするジョブ結合という機能がある。

【0 0 0 3】

【発明が解決しようとする課題】

しかしながら、バナー印刷が設定された複数のジョブを結合し、仕上げ処理やレイアウト処理を指定した場合や、バナー印刷の設定と仕上げ処理やレイアウト処理などの設定が併用された複数のジョブを結合する場合に、結合された 1 つのジョブを指定された設定に従って印刷出力すると、図 9 に示すようなバナー印刷

が設定された複数のジョブを結合してステイプルする場合や2ページ／枚に出力する場合に、ステイプル出力された途中にバナー印刷があったり、バナー印刷と印字データが1ページにレイアウトされたりというように、ユーザが意図しない印刷結果となる場合があった。

#### 【0 0 0 4】

本発明は、上記課題を解決するためになされたものであり、バナー印刷の設定と仕上げ処理やレイアウト処理などの設定が併用されたジョブの結合による不具合を回避することができ、状況に応じて処理を適応することで、ユーザに最適な印刷出力を提供することを目的とする。

#### 【0 0 0 5】

また、これらの処理をバナー印刷、レイアウト処理、仕上げ処理の設定状況に応じて、内部的に自動的に行うことにより、ユーザに意識させることなく最適な出力結果を提供することを目的とする。

#### 【0 0 0 6】

##### 【課題を解決するための手段】

上記目的を達成するために、本発明は、バナー印刷の機能と複数のジョブを結合する機能とを有する印刷システムの印刷制御方法であって、バナー印刷を含む複数の印刷ジョブを結合して所定の加工設定を指定する際に、バナー印刷のデータを複数の印刷データと別に処理する処理工程と、前記複数の印刷データを1つの印刷データに結合する結合工程と、前記処理工程で処理されたバナー印刷のデータと前記結合された印刷データとを出力する出力工程とを有することを特徴とする。

#### 【0 0 0 7】

##### 【発明の実施の形態】

以下、図面を参照しながら本発明に係る実施の形態を詳細に説明する。

#### 【0 0 0 8】

まず、本実施形態を説明する前に、本発明を適用可能なプリンタとプリンタに接続されたパーソナルコンピュータ等の情報処理装置とからなるシステム、特に情報処理装置上で、プリンタに送信する印刷データを生成する前に、一旦、最終

的にプリンタに送付する印刷データとは異なる形式のデータ形式（いわゆる中間コード）での一時保存を行うスプール手段としてのスプーラと、この中間コード形式で一時保存されたデータから改めて最終的にプリンタに送付する印刷データを生成するデスプール手段としてのデスプーラと、プリンタ制御コマンドを生成する手段としてのプリンタドライバとを備える印刷システムの構成について説明する。

#### 【0009】

図1は、本実施形態における印刷システムの構成を説明するブロック図である。尚、本発明に係る機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN、WAN等のネットワークを介して接続がなされ処理が行われるシステムであっても本発明を適用できる。

#### 【0010】

図1に示すように、ホストコンピュータ100は、ROM103のプログラム用ROM或いは外部メモリ111に記憶された文書処理プログラム等に基づいて図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行するCPU101を備え、システムバス104を介して接続される各々のデバイスをCPU101が総括的に制御する。また、このROM103のプログラム用ROM或いは外部メモリ111には、CPU101の制御プログラムであるオペレーティングシステム（以下OS）等が記憶され、ROM103のフォント用ROM或いは外部メモリ111には上述の文書処理の際に使用するフォントデータ等が記憶され、ROM103のデータ用ROM或いは外部メモリ111には上述の文書処理等を行う際に使用する各種データが記憶される。RAM102は、CPU101の主メモリ、ワークエリア等として機能する。

#### 【0011】

キーボードコントローラ（KBC）105は、キーボード（KB）109や不図示のポインティングデバイスからのキー入力を制御する。CRTコントローラ（CRTC）106は、CRTディスプレイ（CRT）110の表示を制御する。ディスクコントローラ（DKC）107は、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、プリンタ制御コ

マンド生成プログラム（以下プリンタドライバ）等を記憶するハードディスク（HD）、フロッピー（登録商標）ディスク（FD）等の外部メモリ111との間のアクセスを制御する。プリンタコントローラ（PRTC）108は、双方向性インターフェース（インターフェース）112を介してプリンタ150に接続され、プリンタ150との通信制御処理を実行する。

#### 【0012】

尚、CPU101は、例えばRAM102上に設定された表示情報RAMへのアウトラインフォントの展開（ラスタライズ）処理を実行し、CRT110上でのWYSIWYGを可能としている。また、CPU101は、CRT110上の不図示のマウ斯卡ーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際に、印刷の設定に関するウィンドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行える。

#### 【0013】

一方、プリンタ150は、CPU151により制御される。このCPU151は、ROM153のプログラム用ROMに記憶された制御プログラム等或いは外部メモリ160に記憶された制御プログラム等に基づいてシステムバス154を介して接続される印刷部（プリンタエンジン）156に出力情報としての画像信号を出力する。また、このROM153のプログラムROMには、CPU151の制御プログラム等が記憶され、ROM153のフォント用ROMには、上述の出力情報を生成する際に使用されるフォントデータ等が記憶され、ROM153のデータ用ROMには、ハードディスク等の外部メモリ160がないプリンタの場合には、ホストコンピュータ上で利用される情報等が記憶されている。

#### 【0014】

CPU151は入力部155を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ100に通知できる。RAM152は、CPU151の主メモリやワークエリア等として機能するメモリで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。尚、RAM152は、出力情報展



開領域、環境データ格納領域、NVRAM等に用いられる。上述したハードディスク（HD）、ICカード等の外部メモリ160は、メモリコントローラ（MC）157によりアクセスを制御される。外部メモリ160は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、操作部159には、操作のためのスイッチ及びLED表示器等が配されている。

#### 【0015】

尚、上述の外部メモリ160は1個に限らず、複数個備えられ、内蔵フォントに加えてオプションカード、言語系の異なるプリンタ制御言語を解釈するためのプログラムを格納した外部メモリを複数接続できるように構成されていても良い。また、図示しないNVRAMを有し、操作部159からのプリンタモード設定情報を記憶するようにしても良い。

#### 【0016】

次に、プリンタ等の印刷装置が直接接続されているか、或いはネットワークを介して接続されているホストコンピュータにおいて実行される典型的な印刷処理について説明する。

#### 【0017】

図2は、ホストコンピュータ100で実行される典型的な印刷処理を示す図である。図中のアプリケーション201、グラフィックエンジン202、プリンタドライバ203、及びシステムスプーラ204は、外部メモリ111に保存されたファイルとして存在し、実行される場合に、OSやそのモジュールを利用するモジュールによってRAM102にロードされ実行されるプログラムモジュールである。また、アプリケーション201及びプリンタドライバ203は、外部メモリ111のFDや不図示のCD-ROM、或いは不図示のネットワークを経由して外部ディスク111のHDに追加することも可能である。

#### 【0018】

まず、印刷を行う場合、外部メモリ111に保存されているアプリケーション201がRAM102にロードされて実行される。このアプリケーション201からプリンタ150に対して印刷を行う旨、キーボード109や不図示のマウス

により指示されると、同様にRAM102にロードされて実行可能となっているグラフィックエンジン202を利用して出力（描画）を行う。

#### 【0019】

ここで、グラフィックエンジン202は印刷装置毎に用意されているプリンタドライバ203を同様に外部メモリ111からRAM102にロードし、アプリケーション201の出力をプリンタドライバ203に設定する。そして、アプリケーション201から受け取るGDI（Graphic Device Interface）関数に基づきDDI（Device Driver Interface）関数に変換し、プリンタドライバ203へDDI関数を出力する。

#### 【0020】

これにより、プリンタドライバ203はグラフィックエンジン202から受け取ったDDI関数に基づき、プリンタ150が認識可能な制御コマンド、例えばPDL（Page Description Language）に変換する。そして、変換されたプリンタ制御コマンドは、OSによってRAM102にロードされたシステムスプーラ204に渡され、インタフェース112経由でプリンタ150へ印刷データとして出力される。

#### 【0021】

次に、本実施形態における印刷システムについて説明する。本実施形態では、図2に示すプリンタとポストコンピュータからなる印刷システムに加えて、更に図3に示すバナー印刷を行うための構成と、図7に示すアプリケーションからの印刷データを一旦中間コードでスプールする構成とを有するものである。

#### 【0022】

図3は、プリンタドライバによるバナー印刷を説明するための図である。尚、バナー印刷の設定は、プリンタドライバ203によって提供される、図4に示すユーザインターフェース301上において、ユーザによって印刷処理の開始前に予め行われる。

#### 【0023】

図4は、バナー印刷を指定するユーザインターフェース301の一例を示す図である。この例では、バナー印刷はチェックボックス401の形式で提供される

。このチェックボックス 401 をチェックすることにより、ユーザはバナー印刷の指定を行い、またプリンタドライバ 203 は、この設定に基づきバナー印刷の処理を行うものである。また、ユーザがバナー編集ボタン 402 を押下することで、バナー設定のポップアップ画面 403 が表示され、ユーザはバナー上に表示させたい情報を選択することができる。図 4 に示す例では、印刷者名、印刷日時、原稿ファイル名、ホスト名（コンピュータ名）をそれぞれ選択できる。また、印刷者名はユーザが任意に指定できるものである。これら設定は RAM 102 上、もしくは外部メモリ 111 上に一時保存され、実際の印刷実行時に、再度プリンタドライバ 203 より呼び出され、印刷処理に反映される。

#### 【0024】

図 3 に戻り、アプリケーション 201 より印刷処理が開始されると、プリンタドライバ 203 はユーザインターフェース 301 上でバナー印刷の設定がされたか否かを判断し、設定がされている場合は、バナー印刷用の印字データを内部的に作成する（302）。作成されたバナー用の印字データは、アプリケーション 201 からの指定に基づき生成された通常の印字データ 303 と併合され、その他の仕上げ処理やレイアウト指定などの印刷設定を指定するコマンド 304 が付加された後、通常印刷時と同様にシステムスプーラ 204 に出力される。

#### 【0025】

図 5 は、システムスプーラ 204 に出力されるデータの一例を示す図である。図 5 に示すように、本実施形態ではデータの構造を階層化しており、第二階層を二つに分割することで、内部的にバナー印刷用の印字データ 501 と、その他のアプリケーション指定の実際の印字データ 502 とを区別し、また仕上げ処理やレイアウト指定といった印刷設定 503 をそれぞれの第二階層に対して指定することで、バナー印刷を通常の印字データに施されたレイアウトや仕上げといった印刷設定と切り離された一枚の単独の紙へ行くことを可能とするものである。

#### 【0026】

これ以降は通常の印刷時と同様にプリンタ 150 へ転送され、プリンタ 150 ではこれら印刷コマンドに従い、バナー印刷、及び通常印刷の出力を行う。

#### 【0027】

図 6 は、本実施形態におけるバナー設定時の出力結果の一例を示す図である。図 6 において、6 0 1 はバナー印刷のみを指定した場合の出力結果である。これに対して、6 0 2 はバナー印刷とステイプルの設定、また 6 0 3 はバナー印刷とレイアウトの変更（2 ページ／枚）を併用した場合の出力結果である。

#### 【0 0 2 8】

ここで、上述のバナー印刷のジョブと、レイアウトや仕上げといった印刷設定のジョブとのジョブ結合を実現する方法について説明する。

#### 【0 0 2 9】

図 7 は、グラフィックエンジン 2 0 2 からプリンタドライバ 2 0 3 へ印刷命令を送る際に、一旦中間コードからなるスプールファイル 7 0 3 を生成する構成を示す図である。図 7 で示すシステムにおいては、スプールファイル 7 0 3 の内容に対して加工することができる。これによりアプリケーション 2 0 1 からの印刷データに対して、拡大縮小や複数ページを 1 ページに縮小して印刷する等、アプリケーション 2 0 1 の持たない機能を実現することができる。

#### 【0 0 3 0】

これらの目的のために、図 2 に示したシステムに対して、図 7 に示すシステムでは中間コードでスプールするようシステムが拡張されている。尚、印刷データの加工を行うためには、通常プリンタドライバ 2 0 3 が提供するウインドウから設定を行い、プリンタドライバ 2 0 3 がその設定内容を R A M 1 0 2 上、或いは外部メモリ 1 1 1 上に保管する。

#### 【0 0 3 1】

次に、図 7 に示すシステムの印刷処理について詳細に説明する。図示するように、この拡張された処理方式では、グラフィックエンジン 2 0 2 からの印刷命令である D D I 関数をデイスパッチャ 7 0 1 が受け取る。デイスパッチャ 7 0 1 がグラフィックエンジン 2 0 2 から受け取った印刷命令（D D I 関数）が、アプリケーション 2 0 1 からグラフィックエンジン 2 0 2 へ発行された印刷命令（G D I 関数）に基づくものである場合には、デイスパッチャ 7 0 1 は外部メモリ 1 1 1 に格納されているスプーラ 7 0 2 を R A M 1 0 2 にロードし、プリンタドライバ 2 0 3 ではなく、スプーラ 7 0 2 へ印刷命令（D D I 関数）を送付する。

**【 0 0 3 2 】**

スプーラ 7 0 2 では受け取った印刷命令を解析し、ページ単位に中間コードに変換してスプールファイル 7 0 3 に出力する。このページ単位に格納されている中間コードのスプールファイルをページ描画ファイル（P D F : Page Description File）と呼ぶ。また、スプーラ 7 0 2 は、プリンタドライバ 2 0 3 に対して設定されている印刷データに関する加工設定（製本印刷、N u p、両面、ステイプル、カラー／モノクロ指定等）をプリンタドライバ 2 0 3 から取得してジョブ単位のファイルとしてスプールファイル 7 0 3 に保存する。この時、ジョブ単位に格納されている設定ファイルをジョブ設定ファイル（簡略して S D F : Spool Description Fileと呼ぶこともある）と呼ぶ。このジョブ設定ファイルについては後述する。

**【 0 0 3 3 】**

尚、スプールファイル 7 0 3 は外部メモリ 1 1 1 上にファイルとして生成するが、R A M 1 0 2 上に生成されても構わない。

**【 0 0 3 4 】**

更に、スプーラ 7 0 2 は外部メモリ 1 1 1 に格納されているスプールファイルマネージャ 7 0 4 を R A M 1 0 2 にロードし、そのスプールファイルマネージャ 7 0 4 に対してスプールファイル 7 0 3 の生成状況を通知する。その後、スプールファイルマネージャ 7 0 4 は、スプールファイル 7 0 3 に保存された印刷データに関する加工設定の内容に従って印刷を行えるか判断する。

**【 0 0 3 5 】**

ここで、スプールファイルマネージャ 7 0 4 がグラフィックエンジン 2 0 2 を利用して印刷を行えると判断した場合には、外部メモリ 1 1 1 に格納されているデスプーラ 7 0 5 を R A M 1 0 2 にロードし、そのデスプーラ 7 0 5 に対してスプールファイル 7 0 3 に記述された中間コードのページ描画ファイルの印刷処理を行うように指示する。これにより、デスプーラ 7 0 5 では、スプールファイル 7 0 3 に含まれる中間コードのページ描画ファイルをスプールファイル 7 0 3 に含まれる加工設定情報を含むジョブ設定ファイルに従って加工し、G D I 関数を再生成し、再度グラフィックエンジン 2 0 2 経由で G D I 関数を出力する。

**【0036】**

一方、デイスパッチャ701が、グラフィックエンジン202から受け取った印刷命令（DDI関数）がデスプーラ705からグラフィックエンジン202へ発行された印刷命令（GDI関数）に基づいたものである場合には、デイスパッチャ701はスプーラ702ではなく、プリンタドライバ203にその印刷命令を送る。これにより、プリンタドライバ203ではグラフィックエンジン202から取得したDDI関数に基づいてページ記述言語等からなるプリンタ制御コマンドを生成し、システムスプーラ204経由でプリンタ150に出力する。

**【0037】**

更に、図7では、これまで説明した拡張システムに加え、プレビューア706及び設定変更エディタ707を配し、プレビュー、印刷設定変更、複数ジョブの結合を可能にした例を示している。

**【0038】**

ここで、印刷プレビュー、印刷設定変更、複数ジョブの結合を行うためには、まずユーザが図8に示すプリンタドライバ203によるユーザインターフェース301において「出力先の指定」を行う手段として機能するプルダウンメニュー801により「編集+プレビュー」を指定する必要がある。

**【0039】**

プリンタドライバ203のプロパティ（図8に示す301）で設定されている内容は設定ファイルとしてOSが提供する構造体（Windows（登録商標）OSでは、DEVMODEと呼ばれる）に格納される。その構造体には、例えばスプールファイル703に含まれる加工設定中に、スプールファイルマネージャ704にストアを行うか否かの設定が含まれており、スプールファイルマネージャ704がプリンタドライバ203を介して加工設定を読み込み、ストア指定がなされていた場合、上述したようにスプールファイル703にページ描画ファイルとジョブ設定ファイルとが生成・格納される。そして、図8に示すように、スプールファイルマネージャのウインドウ画面802がポップアップされ、スプールファイル703にスプールされたジョブがリスト表示される。

**【0040】**

尚、図 8 に示す画面 8 0 2 は、3 つのジョブがスプールされている例を示しており、メニューバーもしくは、そのすぐ下のメニューアイコンを押下することにより、ジョブの操作を行うことができる。また、操作の種類としては、ジョブを選択した状態で「印刷」、中間コードのスプールファイルをそのまま残して印刷を行わせる「セーブして印刷」、印刷設定を考慮したジョブの出力プレビューを見るための「プレビュー」、中間コードのスプールファイルを削除する「削除」、中間コードのスプールファイルのコピーを生成する「複製」、複数の中間コードのスプールファイルのジョブを結合して 1 つのジョブにする「結合」、結合ジョブを元の複数のジョブに分割する「分割」、単体ジョブもしくは結合ジョブの印刷設定（レイアウト設定やフィニッシング設定等）を変更する「ジョブ編集」、あるジョブの印刷順序を最初にする「先頭に移動」、印刷順序を 1 つ前にする「1 つ上に移動」、印刷順序を 1 つ後にする「1 つ下に移動」、印刷順序を最後にする「最後に移動」の以上 1 1 個の操作がある。但し、ここでは、本実施形態と関連する「結合」の操作について説明する。

#### 【0 0 4 1】

まず、ジョブの結合処理は、図 8 に示すように、スプールファイルマネージャのウインドウ画面 8 0 2 に表示されている複数のスプールファイルを選択した後、画面 8 0 2 のメニューバー、もしくは結合を指定するアイコンボタン 8 0 3 を操作することによって開始される。この結合処理が開始された場合、外部メモリ 1 1 1 に格納されている設定変更エディタ 7 0 7 を R A M 1 0 2 にロードし、設定変更エディタ 7 0 7 に対してリスト上の先頭ジョブ、或いはデフォルトの加工設定の表示を行うように指示する。そして図 8 に示すように、設定変更エディタ画面 8 0 4 が表示される。

#### 【0 0 4 2】

この設定変更エディタ画面 8 0 4 では、それぞれのジョブに対して、結合する前の加工設定で表示することも、結合ジョブとして統一の加工設定に変更、修正して表示することも可能である。その際、プリンタドライバ 2 0 3 の設定可能な項目を設定変更エディタ 7 0 7 上のユーザインターフェースに表示しても、プリンタドライバ 2 0 3 自身のユーザインターフェースを呼び出しても構わない。

**【0043】**

ここで、結合されたジョブ及び変更された変更項目は、設定変更エディタ画面 804 上の「結合」ボタン 805 の押下によるユーザの認証要求に従い、変更が認証されると、その制御がスプールファイルマネージャ 704 に移行する。これらの操作により、先に選択された複数ジョブは、スプールファイルマネージャのウィンドウ 806 上で 1 つの結合ジョブ 807 として表示される。

**【0044】**

そして、スプールファイルマネージャ画面 806 上に設けられたメニューバー、もしくは印刷を指定するアイコンボタン 808 を操作することによってユーザが印刷要求を行うならば、スプールファイルマネージャ 704 はその印刷要求を発行する。印刷要求が発行されるとデスプーラ 705 はスプールファイルマネージャ 704 にて指定されている結合の設定に基づき、その結合ジョブに含まれるページの中間データを再度グラフィックエンジン 202 へと出力する。これを受けてグラフィックエンジン 202 からディスパッチャ 701 経由でプリンタドライバ 203 に印刷命令が送られ、印刷が実行される。プリンタドライバ 203 に送られた結合された印刷ジョブは、通常の印刷と同様に 1 つの印刷ジョブとして処理され、印字データに変換された後、システムスプーラ 204 経由でプリンタ 150 に転送される。

**【0045】**

これにより、プリンタ 150 では、ジョブ中に指定された印刷設定に従い印刷を行う。このとき、プリンタ 150 に転送されるジョブは、データ構成上ジョブの結合を意識させないものとなっているので、1 つのジョブとして一括の仕上げ処理、レイアウト処理などを行うことが可能となる。

**【0046】**

以上のバナー印刷、ジョブ結合の二つの機能を持つ印刷システムを前提とする本発明に係る実施形態の一例について以下に説明する。

**【0047】**

本発明が適応されるのは、バナー印刷とジョブ結合が併用される場合である。図 9 は、バナー印刷とジョブ結合が併用された場合の印刷結果の一例を示す図で



ある。バナー印刷を設定された印刷ジョブ 9 0 1 に対してジョブ結合動作を行う場合、ユーザによる印刷設定に忠実な印刷出力を得るためには、各印刷ジョブに設定されたバナー印刷の設定を反映させたままの状態で行い、生成される印刷物が 9 0 2 に示すように、バナー印刷がそれぞれのアプリケーションからの通常印刷の先頭に付与されるように印刷されることが期待される。

#### 【 0 0 4 8 】

しかしながら、このような結合ジョブに対して、仕上げ指定、レイアウト変更などの処理を行うと、バナー印刷も含めた形での処理が行われてしまい、結果としてユーザにとって不都合な印刷物を提供してしまうことになる。図 9 において、9 0 3 はステイプルを指定した場合の出力結果であり、9 0 4 は 2 ページ／枚を指定した場合の出力結果である。この他にも、両面印刷やページ番号を付ける場合などに同様に不具合が予想される。

#### 【 0 0 4 9 】

このような不具合を回避するために、本実施形態では図 1 0 に示す処理形態を取る。上述したように、仕上げ指定やレイアウト変更などの処理とバナー印刷とが指定された複数の印刷ジョブ 1 0 0 1 に対してジョブ結合の処理を行う際に、スプールファイル 7 0 3 におけるバナー印刷の設定 1 0 0 3 を破棄する。次に、バナー設定の取り除かれた通常の印刷ジョブを通常のジョブ結合と同様に処理を行い、一つの印刷ジョブ 1 0 0 2 を生成する。そして、その結合ジョブに対して印刷処理を開始するタイミングで、新たなバナー印刷 1 0 0 4 の設定を内部的に生成し、結合された印刷ジョブ 1 0 0 2 に付加する。これにより、ステイプルなどの仕上げ処理 1 0 0 5 の場合や 2 ページ／枚などのレイアウト変更の処理 1 0 0 6 を行った場合にも、図 1 0 に示すようなユーザにとって望ましい形での出力を得ることができるようになる。

#### 【 0 0 5 0 】

尚、上述の処理は仕上げ処理やレイアウト変更などの設定が併用されるような不具合の予想される場合を対象とするもので、その他の場合に関してはこれらの処理を行わないことにより、今まで通り、ユーザの設定に忠実な印字結果を提供することが可能である。

**【 0 0 5 1 】**

また、本実施形態では、この回避処理への切り替えを自動的に行うことによりユーザの意識しないところで不具合な印刷を回避することを目的としているが、同処理を行うにあたりユーザインターフェース上に機能を持たせることでユーザに明示的に指定させることも可能である。

**【 0 0 5 2 】**

次に、バナー印刷の設定と、仕上げ処理やレイアウト変更などの設定とが併用された複数のジョブの結合処理における詳細について説明する。

**【 0 0 5 3 】**

まず、本実施形態を実現するために、結合処理を行う前の段階としてスプーラ 7 0 2 のスプールファイル 7 0 3 の作成処理を、バナー印刷の設定を考慮して図 1 1 に示すように設定する。そして、上述したユーザインターフェース 3 0 1 及びバナー設定画面 4 0 3 上で指定されたバナー印刷の設定を上述した通りプリンタドライバ 2 0 3 での印字データの生成時に始めて印字データに変換する。即ち、この処理の時点では、設定として R A M 1 0 2 もしくは外部メモリ 1 1 1 に保存されているのみで印字データとなっていない。この設定に基づきスプーラ 7 0 2 は、アプリケーション 2 0 1 から作成される通常印字データのスプールを行うと同時に、内部的に中間データ 1 1 0 1 の作成を行う。これはこの後のジョブ結合処理における操作を容易にするためである。作成された中間データ 1 1 0 1 は、通常の印字データ同様にスプーラ 7 0 2 によってスプールファイル 7 0 3 として保存される。尚、この処理は、バナー印刷が指定されている印刷ジョブに対してのみ行われる。

**【 0 0 5 4 】**

これに対してアプリケーション 2 0 1 からの通常印字データに関してはバナー印刷の設定にかかわらず、通常の処理と同様にグラフィックスエンジン 2 0 2 により中間データ 1 1 0 2 に変換され、スプーラ 7 0 2 によってスプールファイル 7 0 3 として保存される。

**【 0 0 5 5 】**

次に、設定編集エディタ 7 0 7 が実際にジョブの結合を行う際の処理の詳細に

ついて説明する。

#### 【0056】

図12は、ジョブ結合時の処理を示すフローチャートである。まずステップS1201において、ジョブを結合する際にバナー印刷の設定をクリアし、その後結合されたジョブに対してバナーを再設定するためのフラグをオフする。次に、ステップS1202において、結合されたジョブに対して、不具合が予想される仕上げ処理、レイアウト処理などが施されているかを確認する。これらの設定が施されていない場合はステップS1206へ進むが、施されている場合はステップS1203へ進み、バナー印刷が指定されているかを確認する。ここで、バナー印刷の指定がされていない場合はステップS1206へ進むが、指定されていればステップS1204へ進み、上述のフラグをオンにし、続くステップS1205において、スプールファイル703に保存されたバナー設定の中間コードを削除する。

#### 【0057】

次に、ステップS1206において、スプールファイル703上に保存されている印字データに対してジョブの結合処理を開始する。この場合、バナー印刷の設定の有無は特に意識せず、スプールされているジョブに対して通常通りの結合処理を行う。そして、ステップS1207において、結合すべきジョブが残っているか否かを判定し、まだ残っていればステップS1202に戻り、上述の処理を繰り返す。また、指定されたジョブの結合が終了するとステップS1208へ進み、上述のフラグがオンか判定し、オンでなければステップS1207へ進み、バナー設定と仕上げやレイアウト変更などの処理の何れかが指定されていないため、不具合が発生しないのでステップS1211へ進む。

#### 【0058】

また、上述のステップS1208において、フラグがオンであればステップS1209へ進み、内部的に新たにバナー印刷の設定を作成する。次に、ステップS1210において、作成したバナーと、既に結合された印字データとを1つのジョブとする。尚、バナー印刷の設定は、既に1ページの印刷データに相当する中間データとしてスプールされているので、この処理によりユーザ設定に忠実な

位置に自動的に包含される。

#### 【0 0 5 9】

尚、ステップ S 1 2 0 9 で作成するバナー印刷の記載内容は、本実施形態では、プリンタドライバのデフォルトの設定に従うものとするが、新たにユーザインターフェースなどを設けて再度設定し直させるようにしても良く、また結合前のジョブに対して行われている設定のうち 1 つを選択して引き継がせるようにしても良い。

#### 【0 0 6 0】

次に、ステップ S 1 2 1 1 において、結合処理によって作成された結合済みの中間データは、スプールファイルマネージャ 7 0 4 におけるユーザの印刷開始の指定に伴い、デスプーラ 7 0 5、グラフィクスエンジン 2 0 2 を経由してプリンタドライバ 2 0 3 へ送られ、印刷処理にかけられる。

#### 【0 0 6 1】

尚、上述した結合処理で仕上げ処理、レイアウト変更などが指定されなかったジョブの場合、即ちバナーの再設定が行われなかった場合は、既に中間データとして印刷データの 1 ページとしてバナー印刷を包含しているので、このまま通常の印刷処理を行い、印字データの作成、システムスプーラ 2 0 4 への出力を行い、プリンタ 1 5 0 へ転送、出力される。

#### 【0 0 6 2】

図 1 3 は、バナーの再設定が行われなかった場合の印刷ジョブの構造を示す図である。図 5 に示した通常のパナー印刷時と異なり、バナー印刷の印字データが通常印刷データの 1 ページとして存在する形となる。

#### 【0 0 6 3】

これに対して、上述した結合処理で仕上げ処理やレイアウト変更が指定された場合、即ちバナー印刷が再設定されている場合は、その設定に基づき、プリンタドライバ 2 0 3 が通常のパナー印刷の場合と同様に、バナー印刷の印字データを作成する。ここで作成される印字データは結合処理において結合ジョブに対して再度付与された設定に基づくものである。

#### 【0 0 6 4】

図 1 4 は、バナーの再設定が行われた場合の印刷ジョブの構造を示す図である。図 5 に示した通常のパナー印刷時と同様に、バナー印刷が内部的に切り離されており、結合ジョブに対して設定された仕上げ、レイアウトの指定の影響を受けないようになっている。

#### 【 0 0 6 5 】

以上説明した本実施形態によれば、2つの処理形態を状況に応じて使い分けることにより、不具合を回避し、最適な出力をユーザに提供することを可能としている。

#### 【 0 0 6 6 】

尚、本発明は複数の機器（例えば、ホストコンピュータ、インターフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、1つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用しても良い。

#### 【 0 0 6 7 】

また、本発明の目的は前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（CPU若しくはMPU）が記録媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

#### 【 0 0 6 8 】

この場合、記録媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記録媒体は本発明を構成することになる。

#### 【 0 0 6 9 】

このプログラムコードを供給するための記録媒体としては、例えばフロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

#### 【 0 0 7 0 】

また、コンピュータが読み出したプログラムコードを実行することにより、前述

した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0 0 7 1】

更に、記録媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0 0 7 2】

##### 【発明の効果】

以上説明したように、本発明によれば、バナー印刷の設定と仕上げ処理やレイアウト処理などの設定が併用されたジョブの結合による不具合を回避することができ、状況に応じて処理を適応することで、ユーザに最適な印刷出力を提供することが可能となる。

##### 【図面の簡単な説明】

##### 【図 1】

本実施形態における印刷システムの構成を説明するブロック図である。

##### 【図 2】

ホストコンピュータ 1 0 0 で実行される典型的な印刷処理を示す図である。

##### 【図 3】

プリンタドライバによるバナー印刷を説明するための図である。

##### 【図 4】

バナー印刷を指定するユーザインターフェース 3 0 1 の一例を示す図である。

##### 【図 5】

システムスプーラ 2 0 4 に出力されるデータの一例を示す図である。

##### 【図 6】

本実施形態におけるバナー設定時の出力結果の一例を示す図である。

【図 7】

グラフィックエンジン 2 0 2 からプリンタドライバ 2 0 3 へ印刷命令を送る際に、一旦中間コードからなるスプールファイル 7 0 3 を生成する構成を示す図である。

【図 8】

プリンタドライバ 2 0 3 によるユーザインターフェース 3 0 1 を示す図である。

【図 9】

バナー印刷とジョブ結合が併用された場合の印刷結果の一例を示す図である。

【図 1 0】

本実施形態におけるジョブの結合処理を説明するための図である。

【図 1 1】

本実施形態におけるバナーの中間コードと印字データの中間コードとを示す図である。

【図 1 2】

本実施形態におけるジョブ結合時の処理を示すフローチャートである。

【図 1 3】

バナーの再設定が行われなかった場合の印刷ジョブの構造を示す図である。

【図 1 4】

バナーの再設定が行われた場合の印刷ジョブの構造を示す図である。

【符号の説明】

- 1 0 0    ホストコンピュータ
- 1 0 1    C P U
- 1 0 2    R A M
- 1 1 1    外部メモリ
- 1 5 0    プリンタ
- 2 0 1    アプリケーション
- 2 0 2    グラフィックエンジン

2 0 3 プリンタドライバ

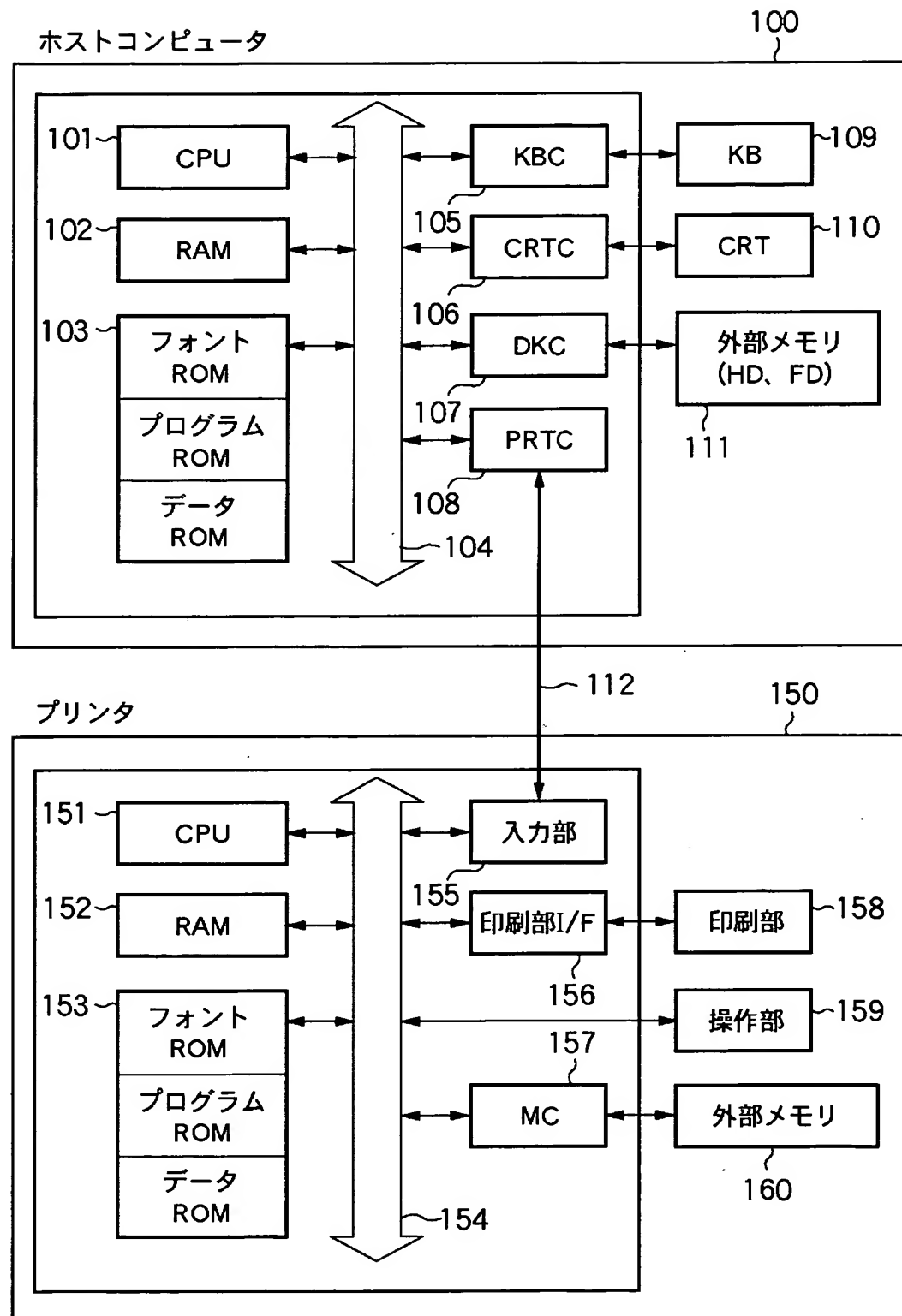
2 0 4 システムスプーラ

3 0 1 プリンタドライバユーザインタフェース

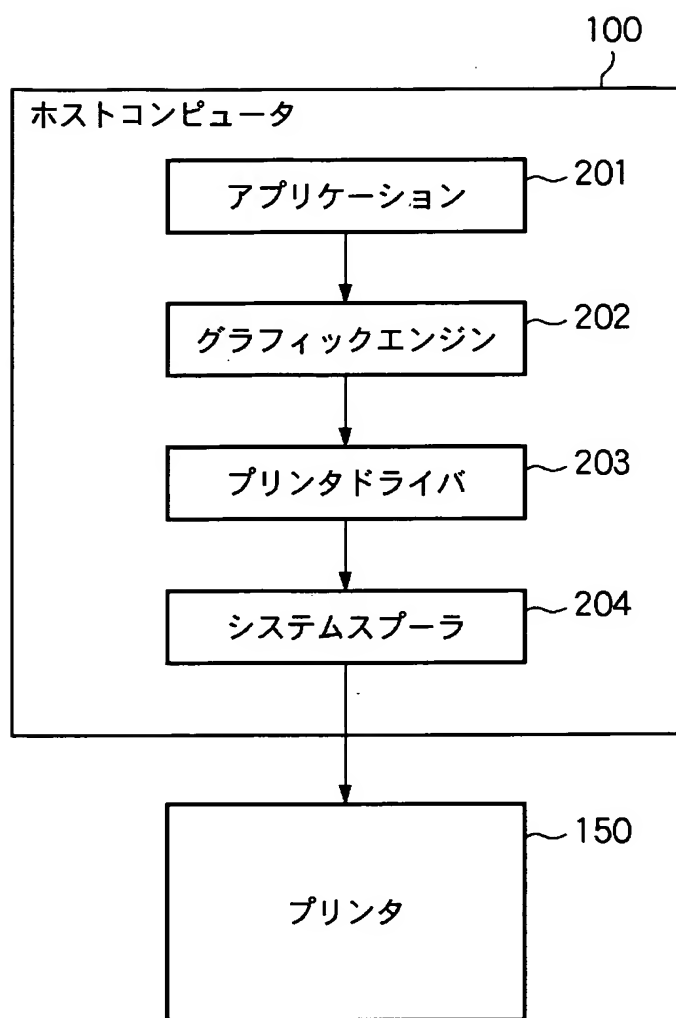


【書類名】 図面

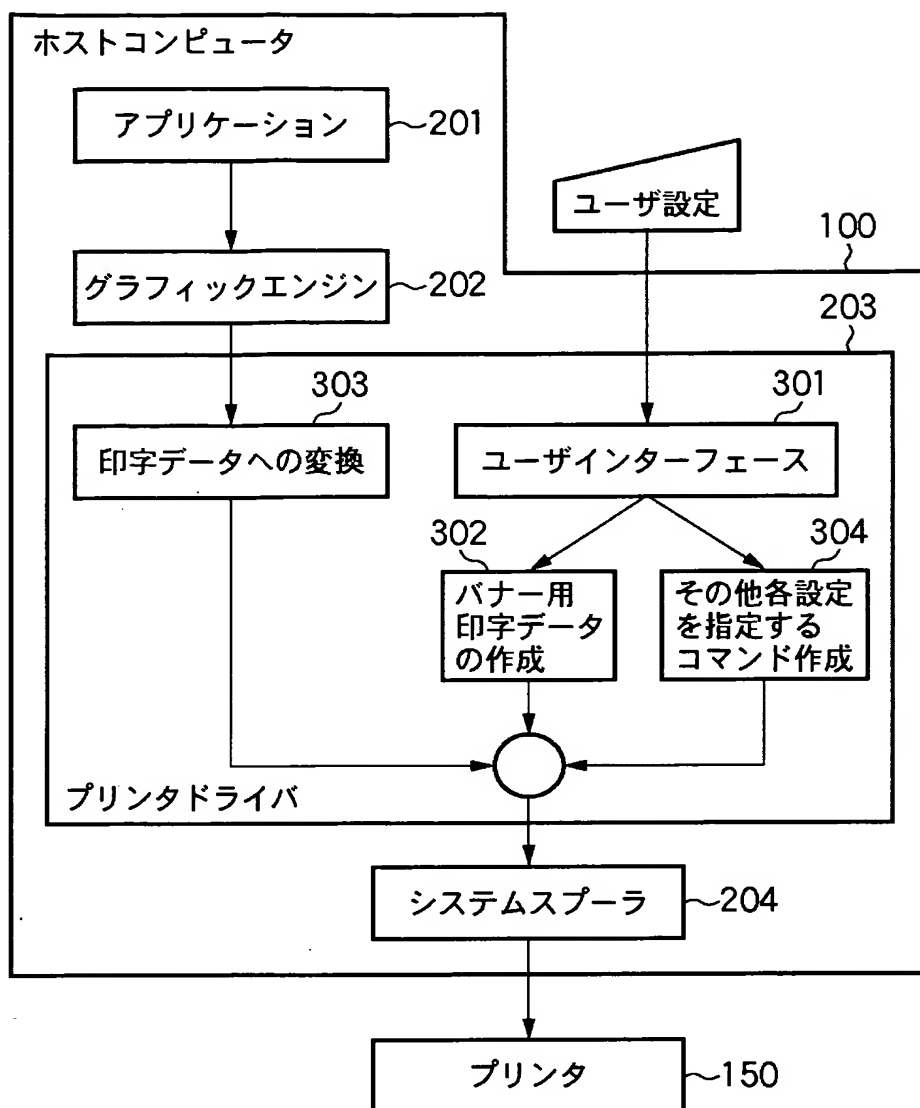
【図 1】



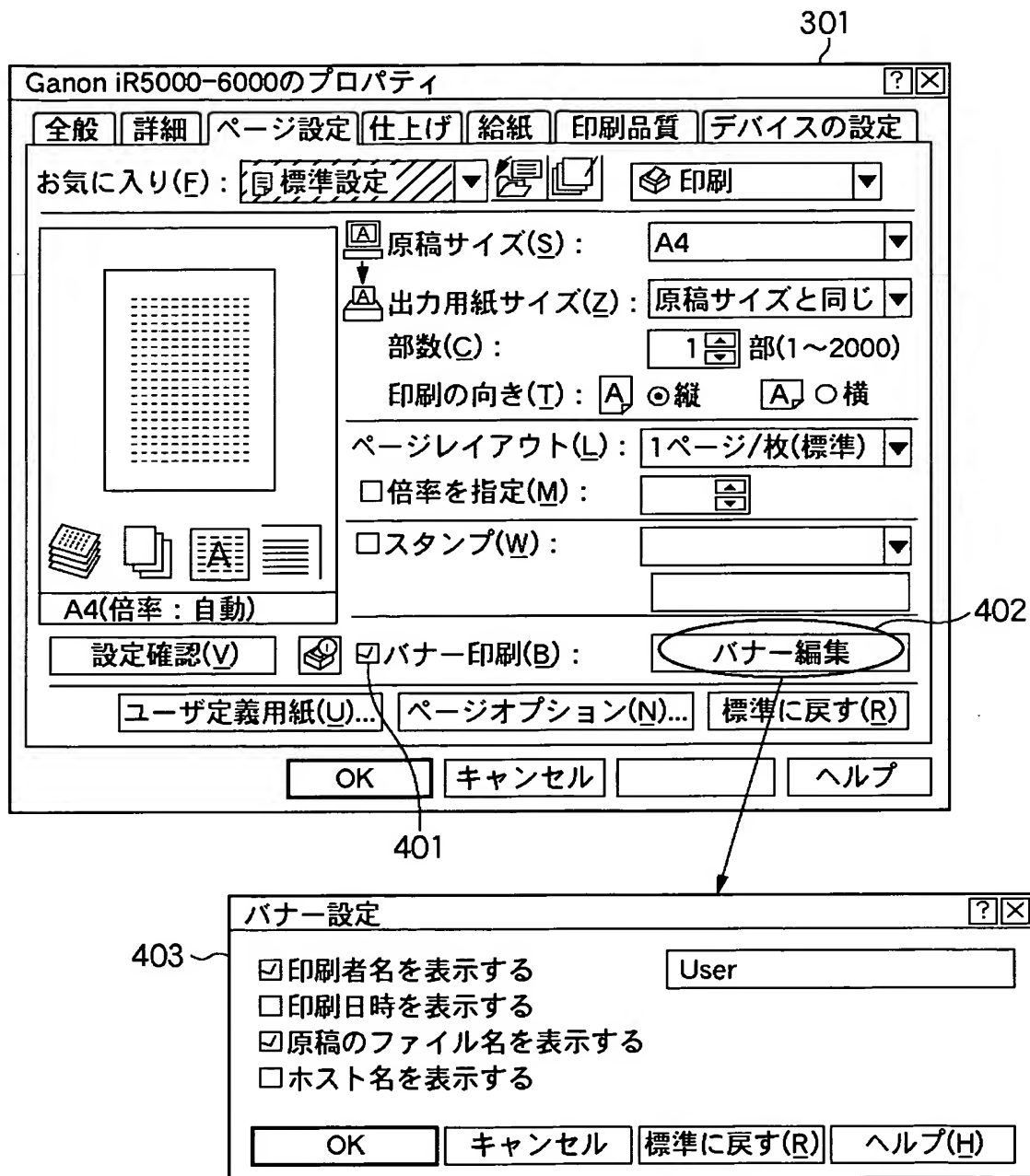
【図 2】



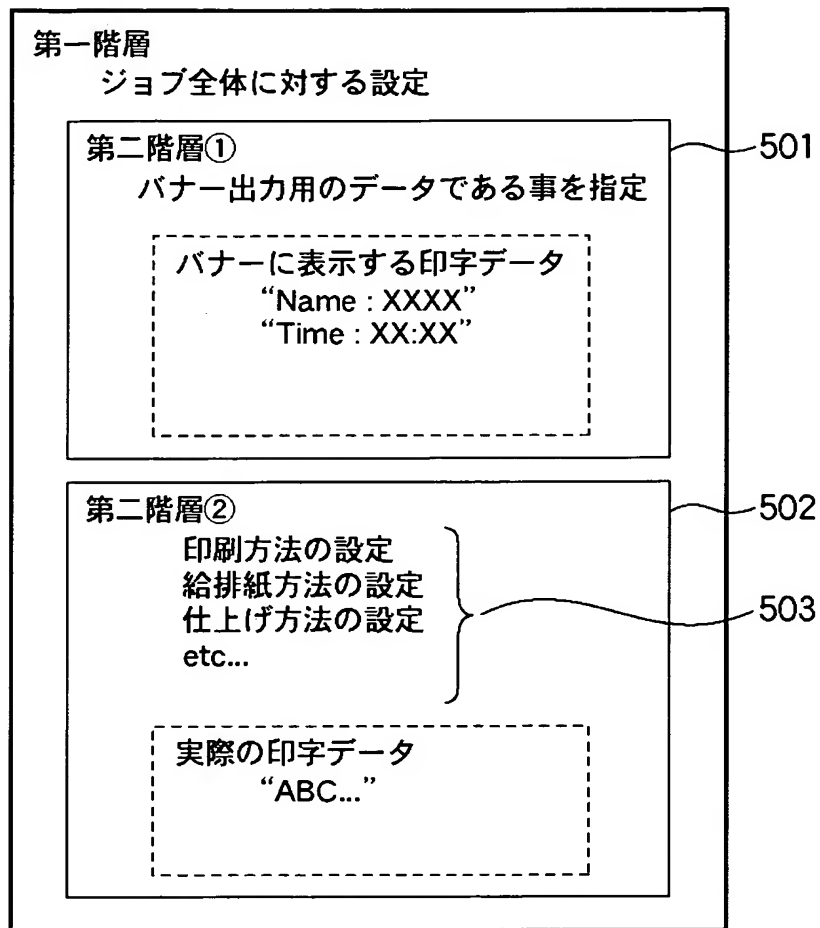
【図 3】



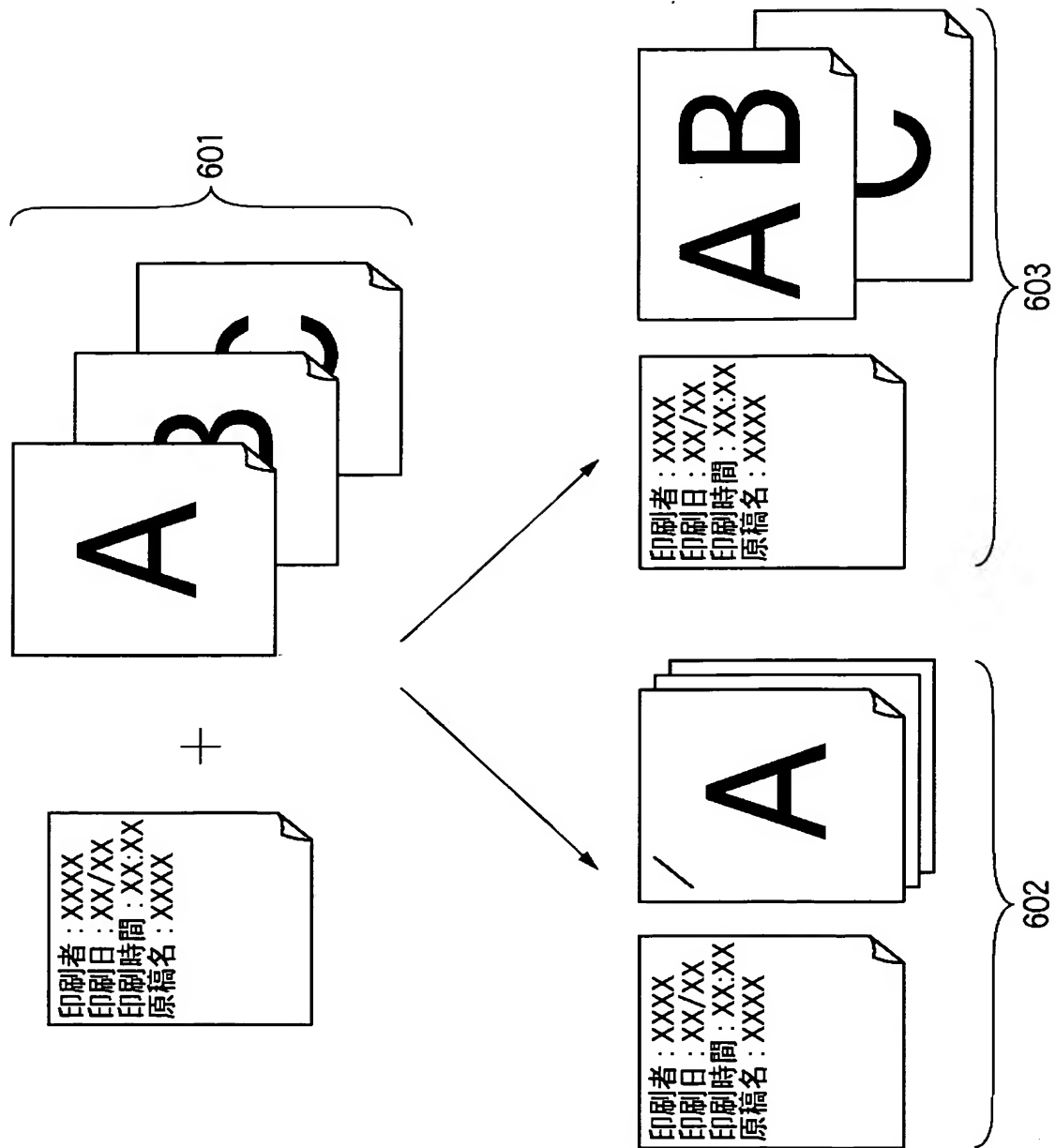
【図 4】



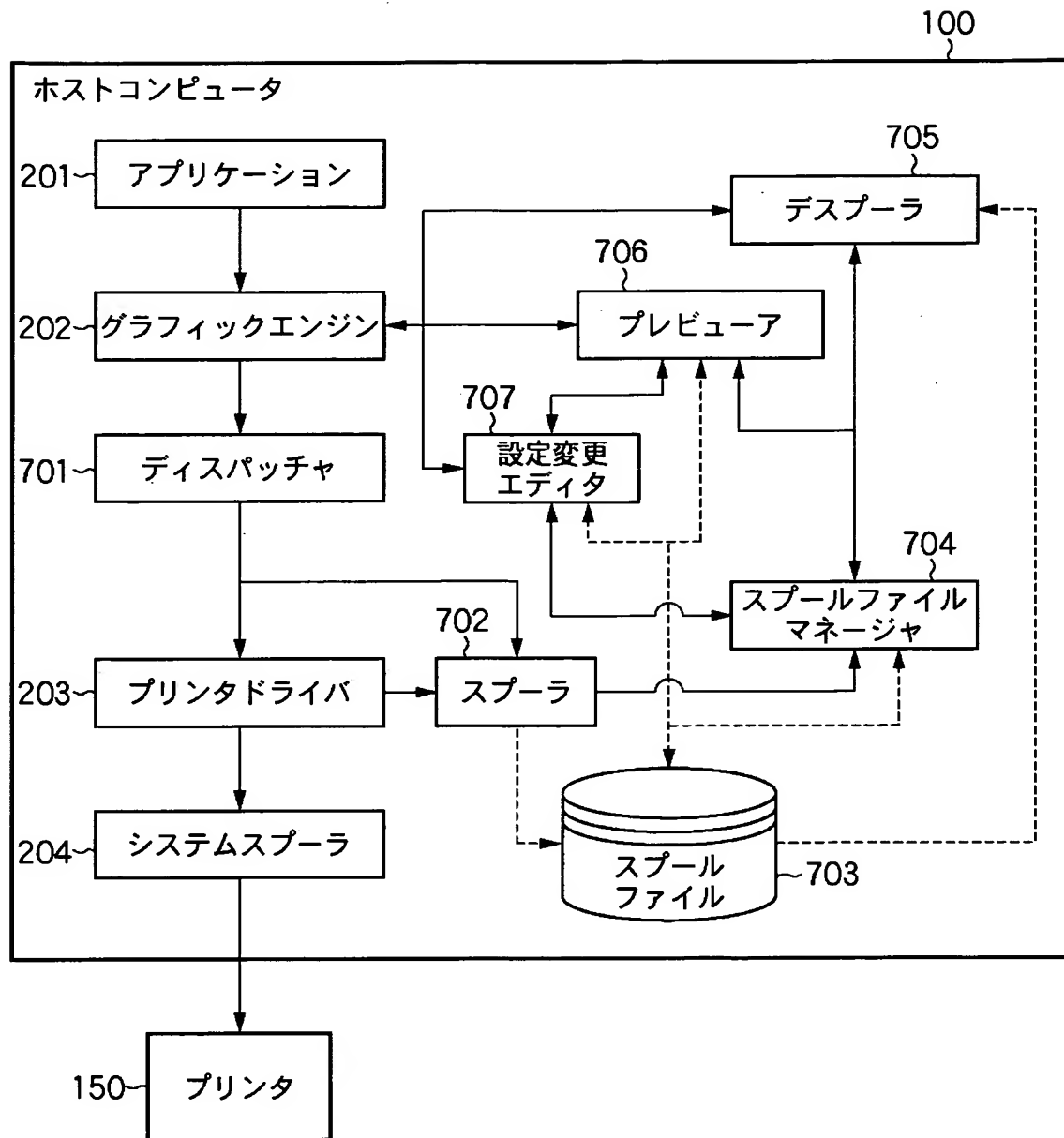
【図 5】



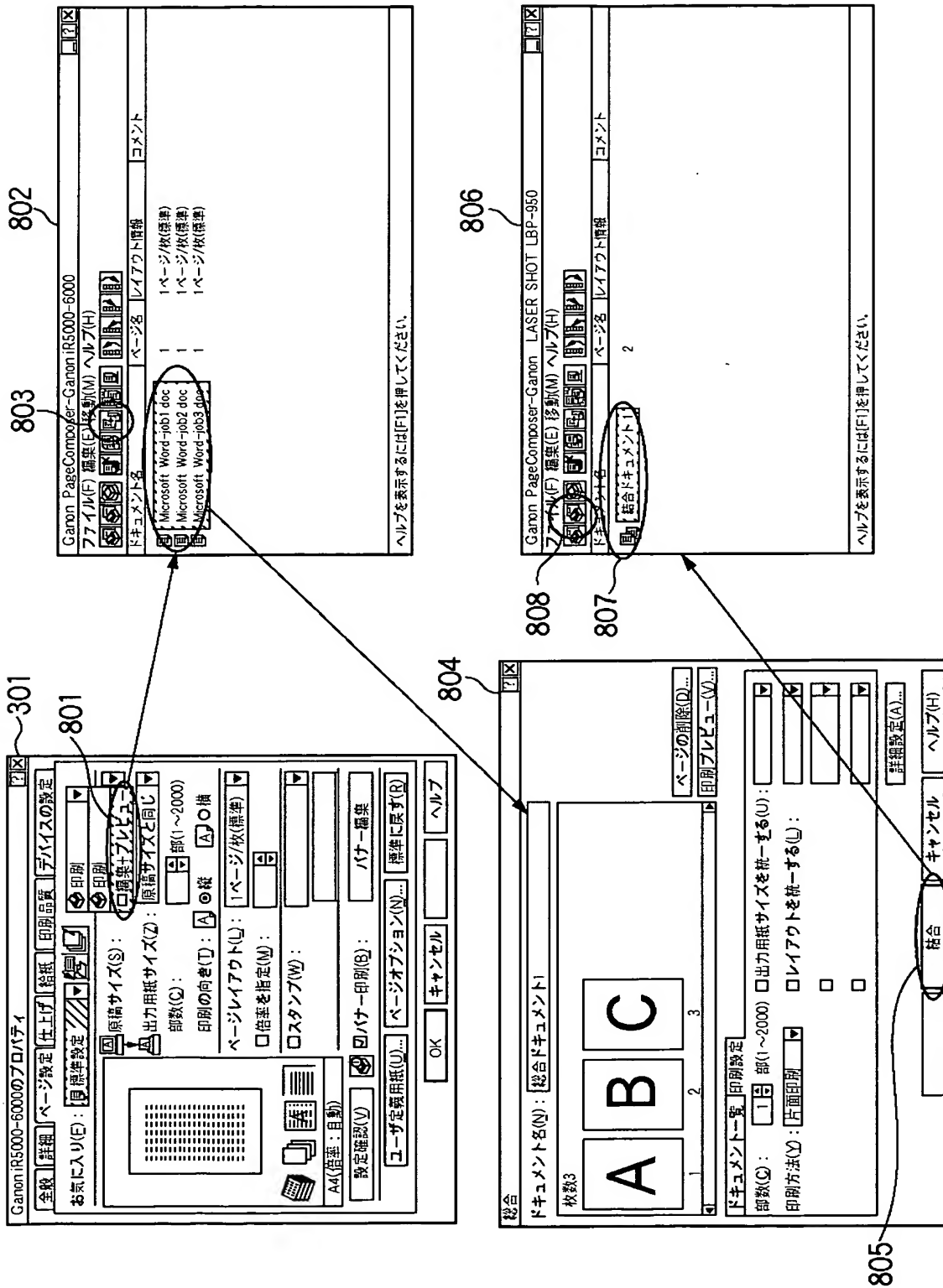
【図 6】



【図 7】

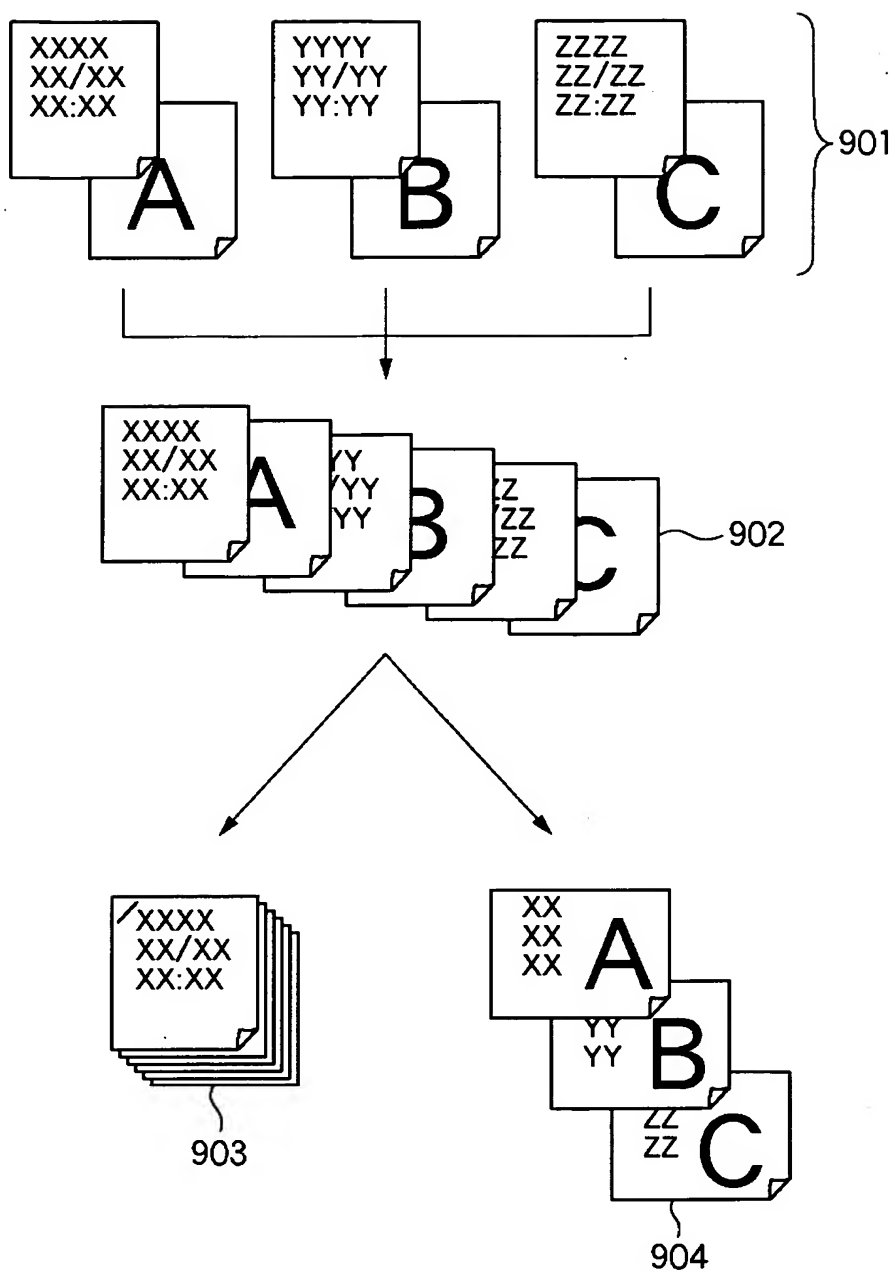


【図 8】

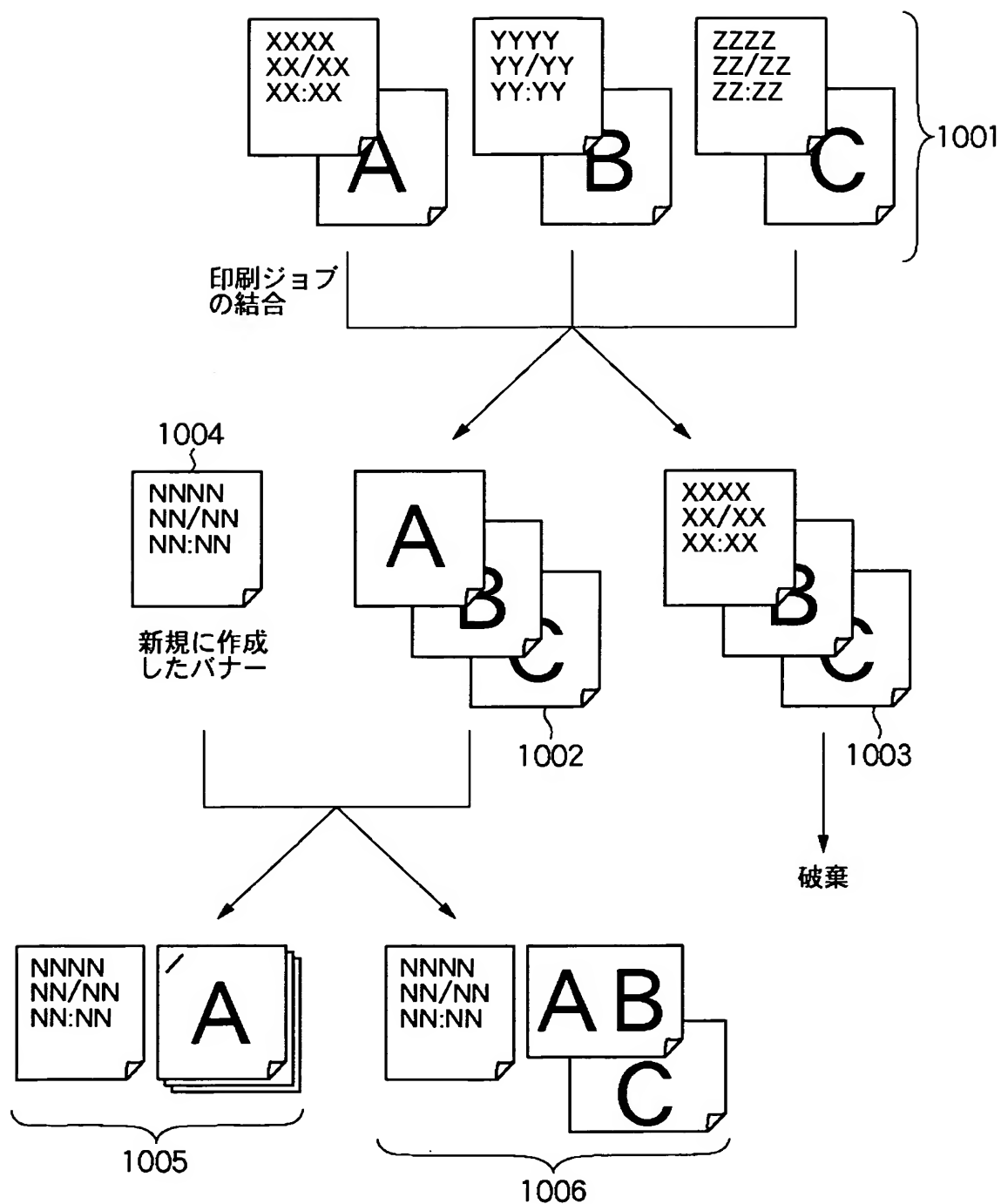




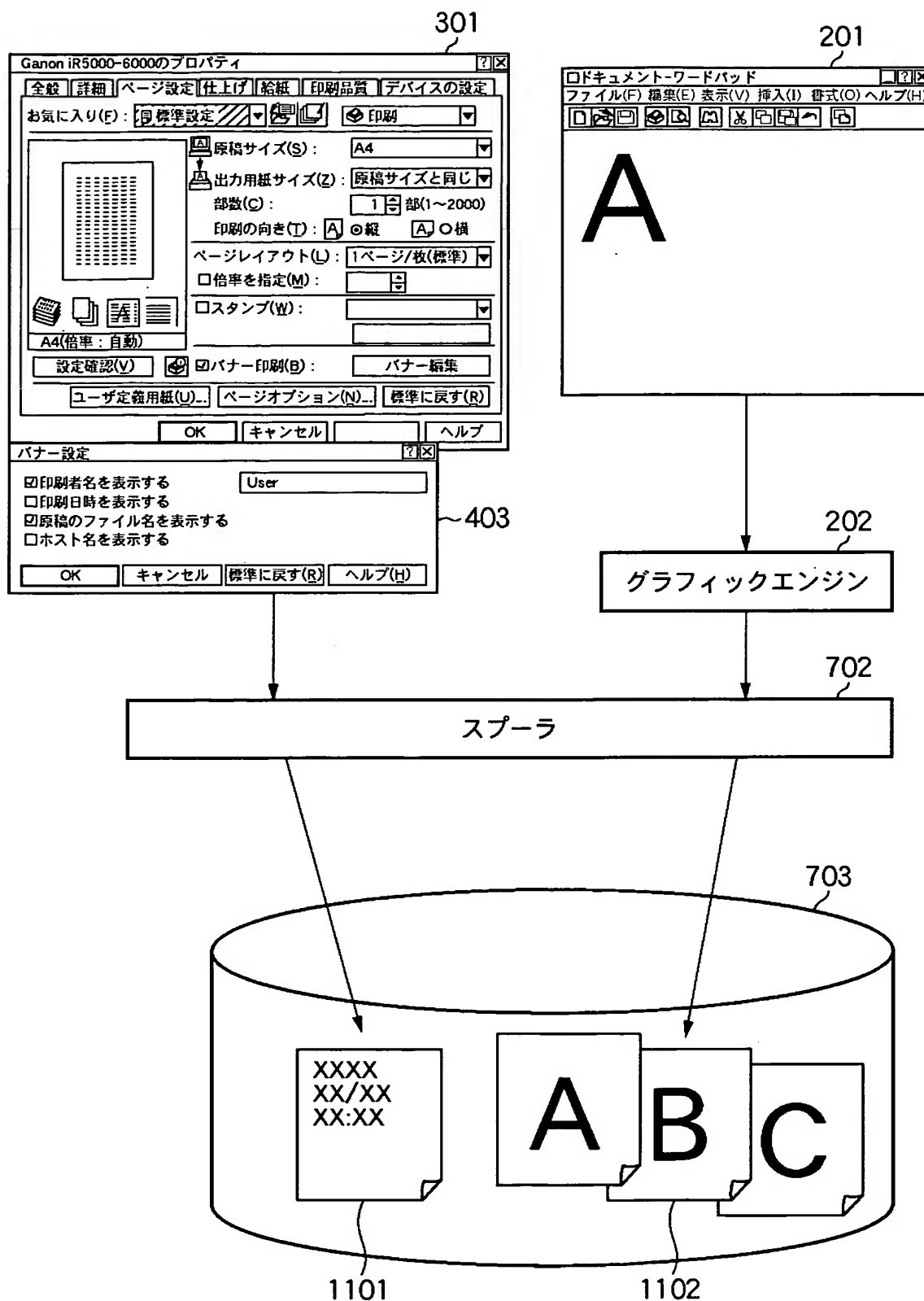
【図 9】



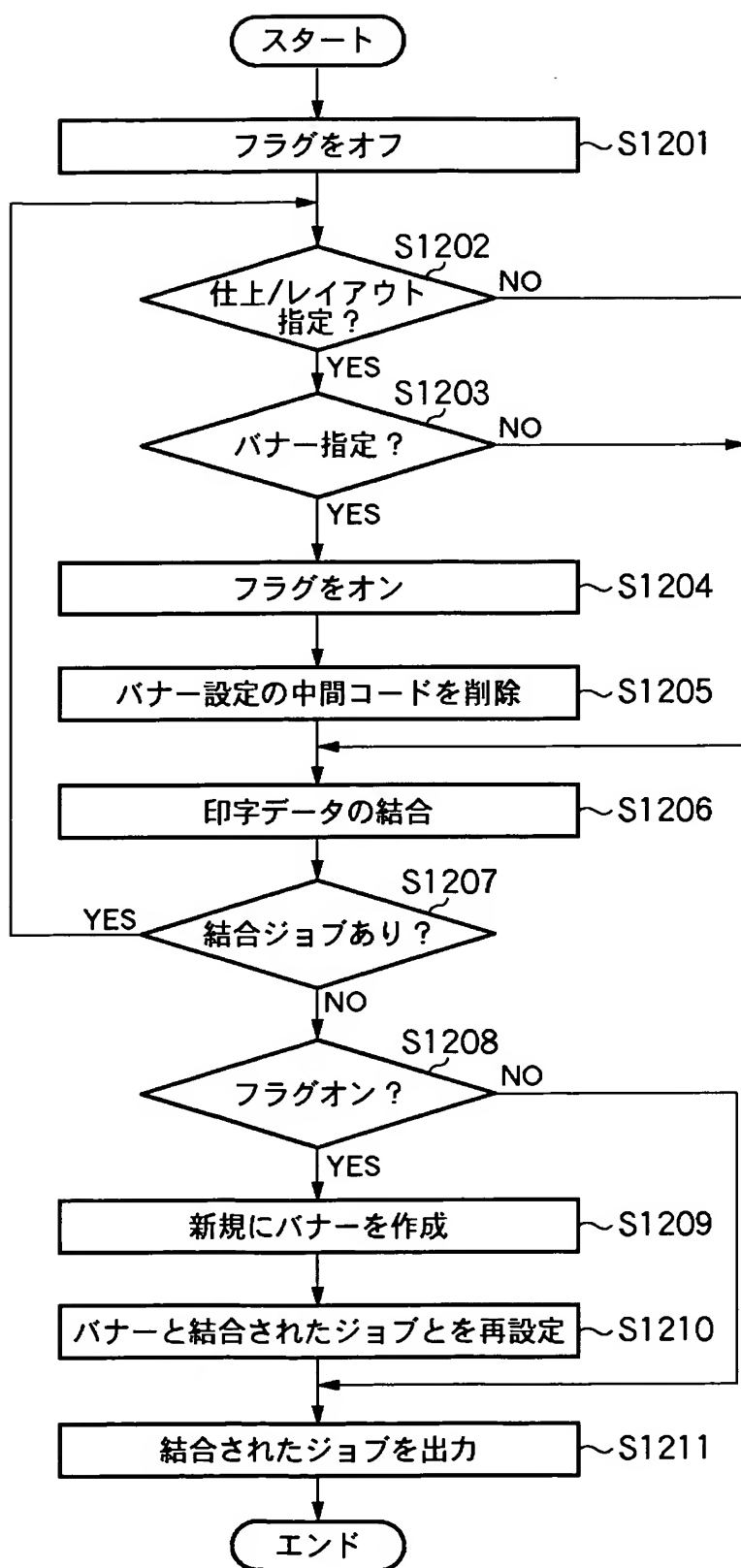
【図 10】



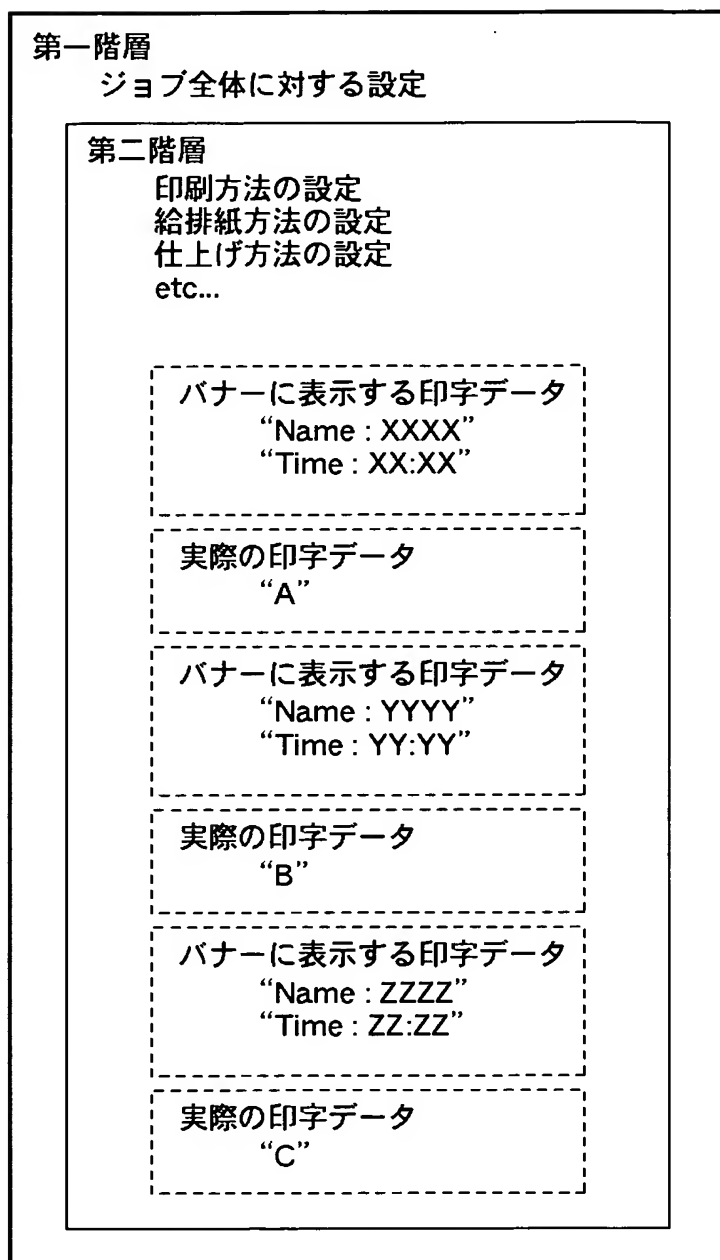
【図 11】



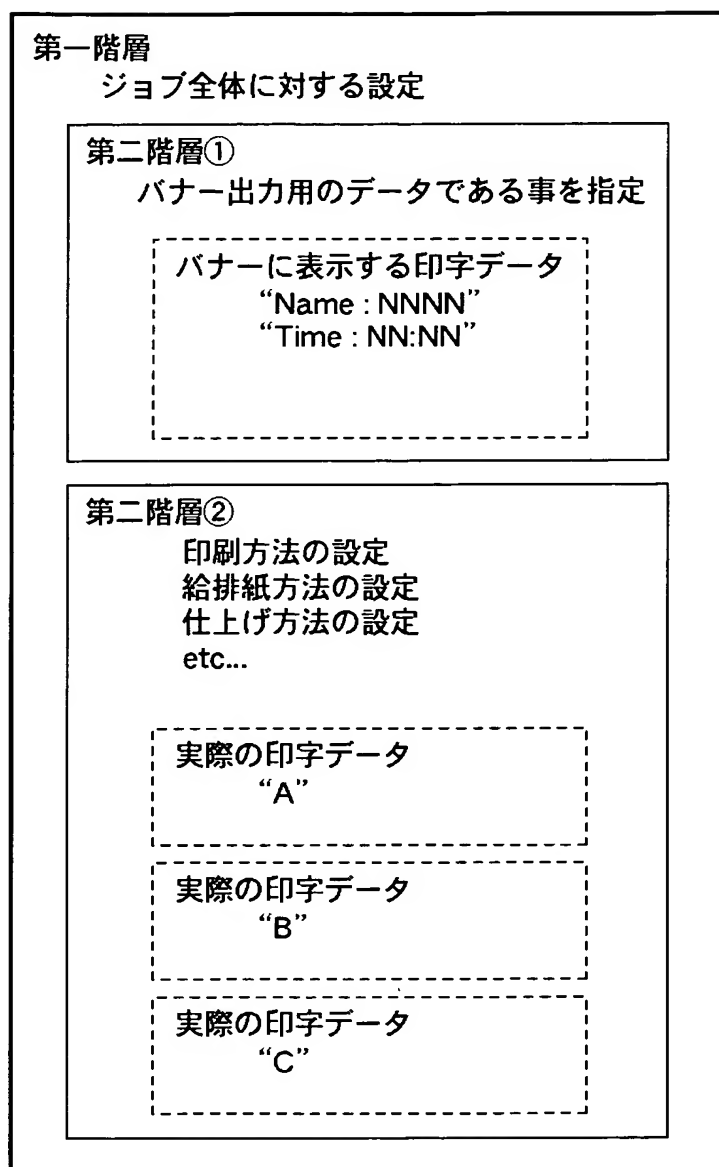
【図 12】



【図 13】



【図 1 4】



【書類名】 要約書

【要約】

【課題】 バナー印刷の設定と仕上げ処理やレイアウト処理などの設定が併用されたジョブの結合による不具合を回避することができ、状況に応じて処理を適応することで、ユーザに最適な印刷出力を提供すること。

【解決手段】 バナー印刷を含む複数の印刷ジョブを結合して所定の加工設定を指定する際に、バナー印刷のデータを複数の印刷データと別に処理（1 0 0 3）し、複数の印刷データを1つの印刷データに結合（1 0 0 2）し、新規に作成したバナー印刷のデータ（1 0 0 4）と、結合された印刷データとを出力する。

【選択図】 図 1 0

特願 2 0 0 2 - 3 5 8 3 1 3

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 1 0 0 7 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キヤノン株式会社